Html Canvas
게임 구현하기

학번 : 2018775050
이름 : 이인석

목차

- 원본 프로그램 설명
- 추가할 기능 설명
- 플레이 영상
- 참고

원본 프로그램 설명
- Style, Canvas, 변수 코드 설명

style 코드

Canvas 태그의 배경색은

색 코드번호 #eee로 설정합니다.

```
<style>
    canvas {
        background: #eee;
    };
</style>
```

canvas 코드

Canvas 태그의 id는 myCanvas로 설정하고

크기는 가로 480 세로 320으로 설정합니다.

```
<canvas id="myCanvas" width="480" height="320"></canvas>
```

변수 코드

Canvas 변수에 id가 myCanvas인 태그의 정보를 넣습니다.
Ctx는 canvas를 2D로 설정합니다.
ballRadius는 공의 크기 변수이고 x, y는 캔버스에서의

공의 좌표 값을 가지는 변수이고, dx, dy는 공의 좌표 이동 속도 값입니다.

```
var canvas = document.getElementById("myCanvas");
var ctx = canvas.getContext("2d");
var ballRadius = 10;
var x = canvas.width/2;
var y = canvas.height-30;
var dx = 2;
var dy = -2;
```

원본 프로그램 설명
- DrawBall, Draw, SetInterval 코드 설명

drawBall 코드

Canvas에서 사용할 공을 만드는 함수입니다.

draw 코드

ClearRect로 캔버스 화면을 지우고

Canvas에서 사용할 공을 생성하고

공이 벽에 충돌하면 팅기게 만드는 함수입니다.

setInterval 코드

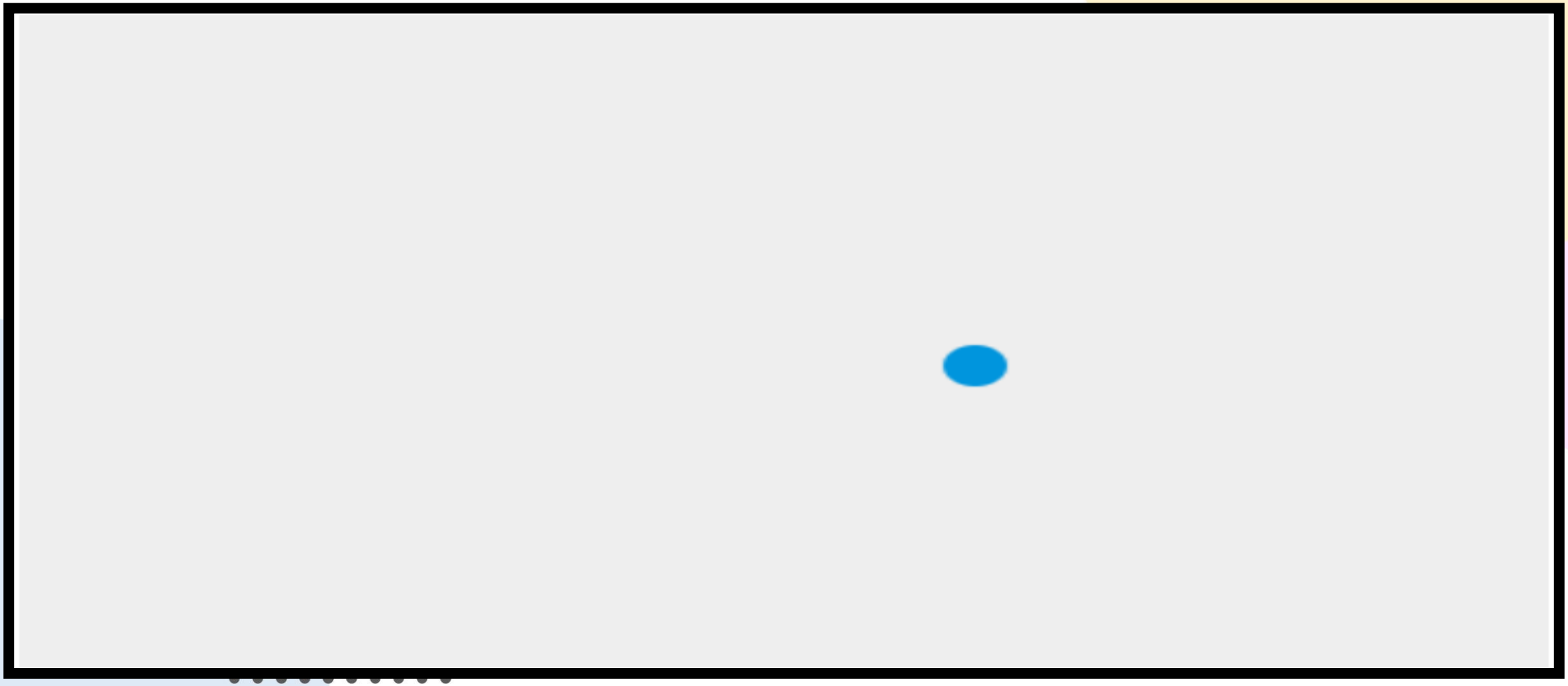
Canvas를 애니메이션 효과를 넣을 함수입니다.

```
function drawBall() {
    ctx.beginPath();
    ctx.arc(x, y, ballRadius, 0, Math.PI*2);
    ctx.fillStyle = "#0095DD";
    ctx.fill();
    ctx.closePath();
}
```

```
function draw() {
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    drawBall();

    if(x + dx > canvas.width-ballRadius || x + dx < ballRadius) {
        dx = -dx;
    }
    if(y + dy > canvas.height-ballRadius || y + dy < ballRadius) {
        dy = -dy;
    }

    x += dx;
    y += dy;
}
setInterval(draw, 10);
```

원본 프로그램 설명

- 실행 화면



추가할 기능설명
- Head 코드 설명(Title, Style)

Head부분 코드

Title 태그로 Html 타이틀을 변경하고

Style 태그로 H1, div, canvas 태그에 position을

left, top, right, bottom 값으로

전 화면을 아무런 제약 없이 쓸 수 있는 Absolute로 설정하고

Transform에서 translate로 x 값을 -50%로 설정하여

화면을 가운데로 정렬시키게 설정합니다.

Div 태그는 추가로 텍스트 타입은 가운데 정렬되고

글 색은 검은색 글은 두껍게 설정합니다.

Canvas 태그는 배경색은 기존 코드 그대로

색 코드 #eee로 설정합니다.

```
<head>
    <title>오래 공 튀기기 게임</title>
    <meta charset="UTF-8">
    <style>

        h1{
            position: absolute;
            top : 5%;
            left: 50%;
            transform: translate(-50%,0%);
        }

        div{
            position: absolute;
            bottom: 40%;
            left: 50%;
            transform: translate(-50%,0%);
            text-align: center;
            color: black;
            font-weight: bold;

        }

        canvas {
            background: #eee;
            position: absolute;
            top : 5%;
            left: 50%;
            transform: translate(-50%,0%);
        }

    </style>
</head>
```

추가할 기능설명
- Body 코드 설명(Canvas, H1, Div)

Body부분 코드

Canvas태그에서 id는 GamePlay

크기는 가로 1920,세로 1080로 설정합니다.

H1태그는 게임점수를 표현하는 태그로

Id는 score 스타일에 display에 값을 none으로 설정하여

맨처음에 화면에 나타나지 않게 만들고 onload event로

게임 시작 시 게임화면에 display되게 설정합니다.

Div태그는 id menu 설정합니다.

```
<body>
    <canvas id="GamePlay" width="1920" height="1080"></canvas>

    <h1 onload="startscore()" id="score" style="display: none" >
        0
    </h1>

    <div id="menu">
        <p style="font-size :40px; margin: 10px;">Mode</p>
        <form >
            <select onchange="modevalue(this)">
                <option>모드 선택</option>
                <option value="T">Time</option>
                <option value="H">Hit</option>
            </select>
        </form>

        <P>Ball Speed <input id="BallSpeed" type="number" max="5" min="2" placeholder="2"></P>
        <P>Ball Size <input id="Ballsize" type="number" max="20" min="10" placeholder="20"></P>
        <P>ball count <input id="balls" type="number" max="3" min="1" placeholder="1"></P>
        <button id="setting" style="padding: 10px 40px; font-size: 20px;">setting</button>
        <br>
        <br>
        <button id="start"style="padding: 12px 50px; font-size: 20px;">start</button>
    </div>
</body>
```

추가할 기능설명
- Body 코드 설명(P, From, Button, Br)

Body부분 코드

첫 번째 P 태그는 글 크기 40px margin을 10px로 설정합니다.

Form 태그는 게임 스코어를 시간, 충돌 중에 선택하기 위해

Onchage event로 modevalue 함수를 호출하게 설정합니다.

두 번째 P 태그는 공의 속도를2~5까지의 범위 id는 Ballspeed 설정합니다.

세 번째 P 태그는 공의 크기를 10~20까지의 범위 id는 Ballsize 설정합니다.

네 번째 P 태그는 공의 개수를 1~3까지의 범위 id는 balls 설정합니다.

첫 번째 Button 태그는 글 크기는 20px padding은 10px, 40px

id는 setting 설정합니다.

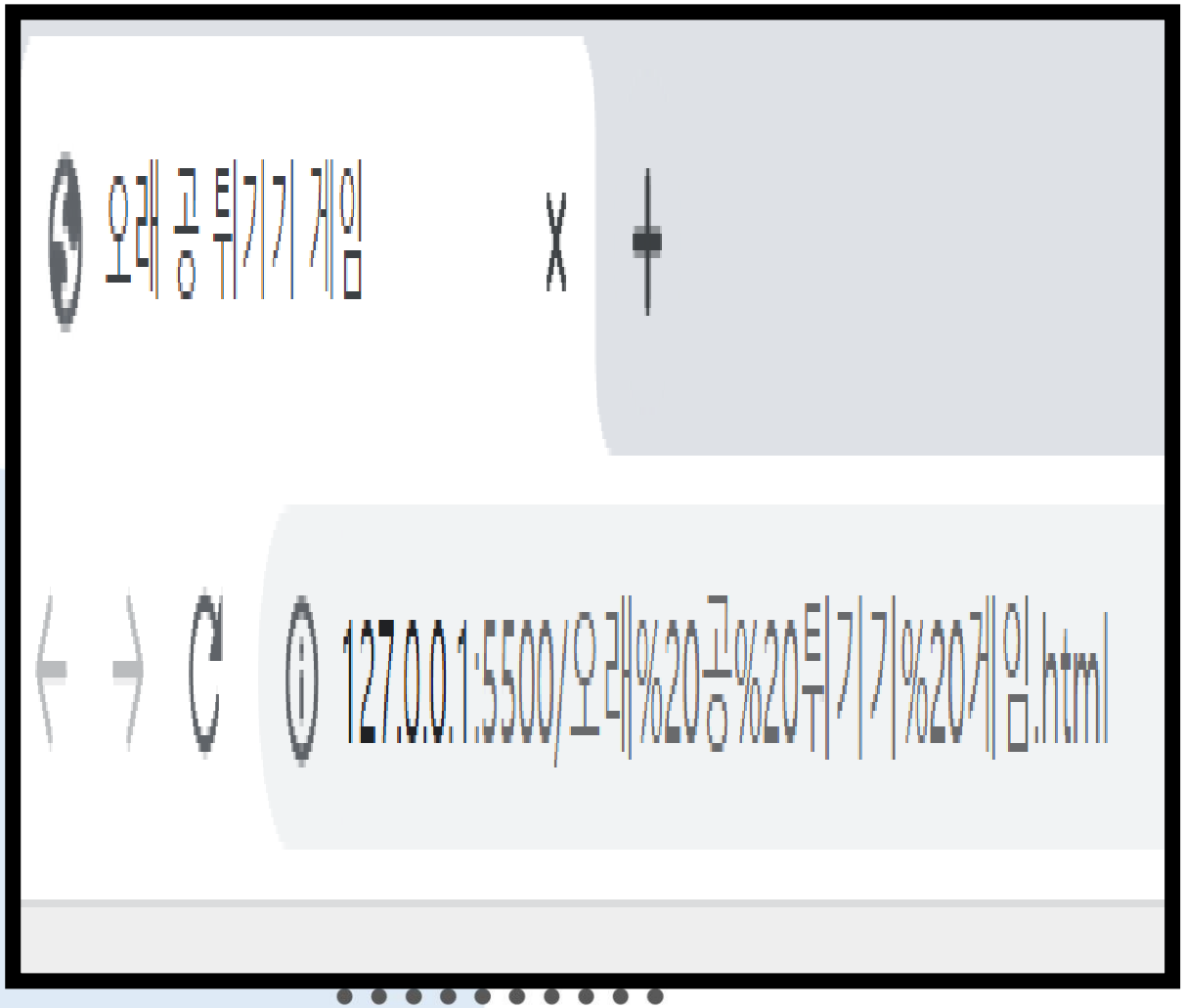
Br 태그는 줄 바꿈 태그로 버튼간 의 거리를 조절합니다.

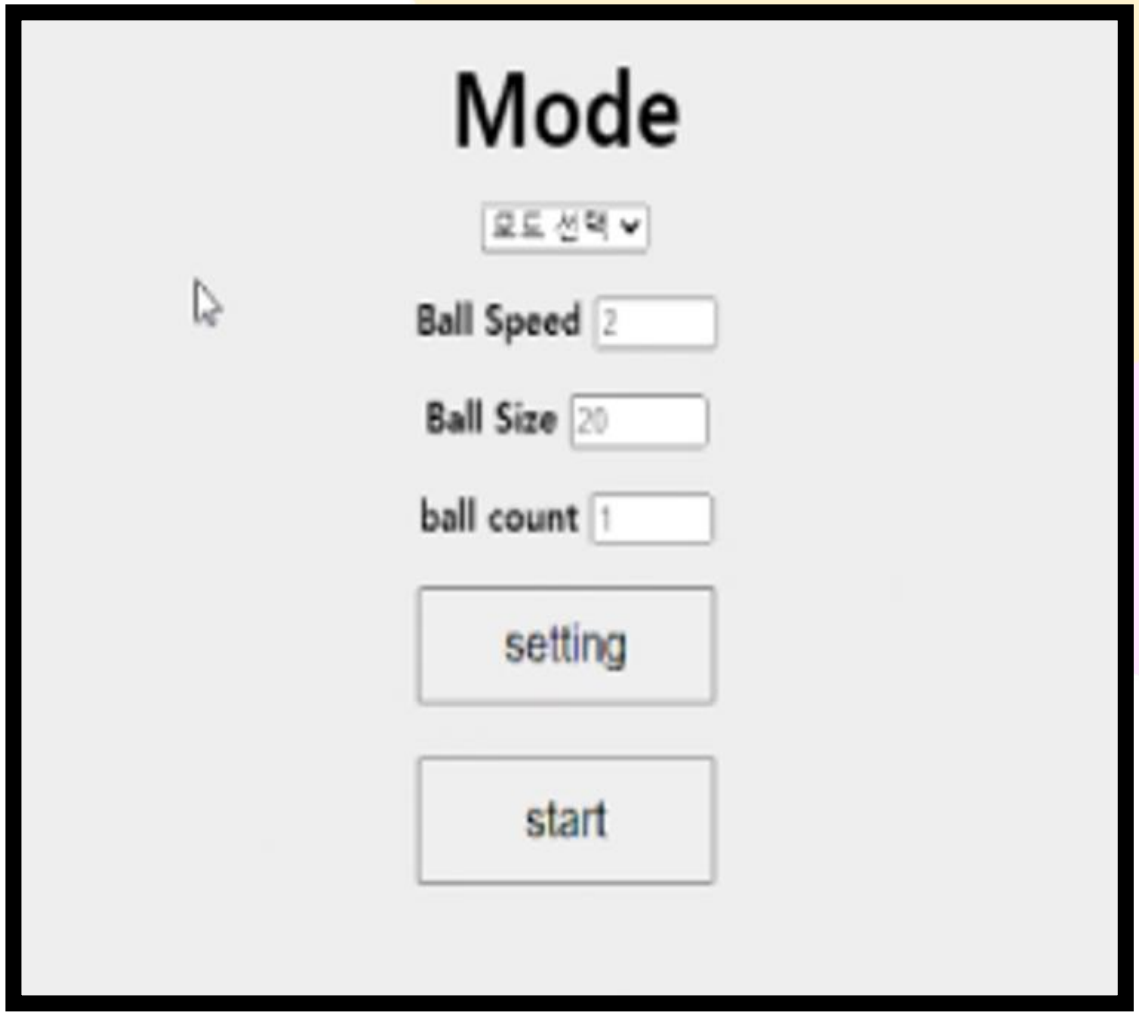
두 번째 Button 태그는 글 크기는 20px padding은 12px, 50px

id는 Start 설정합니다.

```
<body>
    <canvas id="GamePlay" width="1920" height="1080"></canvas>

    <h1 onload="startscore()" id="score" style="display: none" >
        0
    </h1>

    <div id="menu">
        <p style="font-size :40px; margin: 10px;">Mode</p>
        <form >
            <select onchange="modevalue(this)">
                <option>모드 선택</option>
                <option value="T">Time</option>
                <option value="H">Hit</option>
            </select>
        </form>

        <P>Ball Speed <input id="BallSpeed" type="number" max="5" min="2" placeholder="2"></P>
        <P>Ball Size <input id="Ballsize" type="number" max="20" min="10" placeholder="20"></P>
        <P>ball count <input id="balls" type="number" max="3" min="1" placeholder="1"></P>
        <button id="setting" style="padding: 10px 40px; font-size: 20px;">setting</button>
        <br>
        <br>
        <button id="start"style="padding: 12px 50px; font-size: 20px;">start</button>
    </div>
</body>
```

추가할 기능설명
- 실행 화면(Title, Div안의 태그)





추가할 기능설명
- Script 코드 설명(Canvas 변수, Menu 변수)

SCript부분 코드

첫 번째 그림은 Canvas 설정 변수입니다.

두 번째 그림은 메뉴 canvas에서 사용할 변수입니다.

```
// 캔버스 설정 변수
const canvas = document.getElementById("GamePlay");
const ctx = canvas.getContext("2d")
let HitEnd = false // 점수 선택 변수
let tmp // 게임 애니메이션 효과 변수
let tmp1 // 메뉴 애니메이션 효과 변수
let tmp2 // 끝 애니메이션 효과 변수
```

```
// 메뉴 볼 크기 변수
let menuballr1 = 40
let menuballr2 = 40
let menuballr3 = 40
let menuballr4 = 40
let menuballr5 = 40
// 메뉴 볼 좌표 및 속도
let menux1 = canvas.width / 2 // 공의 좌표 x
let menuy1 = canvas.height - 100 // 공의 좌표 y
let menudx1 = 7 // x속도
let menudy1 = -7// y 속도

let menux2 = canvas.width / 2 + 200 // 공의 좌표 x
let menuy2 = canvas.height - 200 // 공의 좌표 y
let menudx2 = 7 // x속도
let menudy2 = -7// y 속도

let menux3 = canvas.width / 2 + 400// 공의 좌표 x
let menuy3 = canvas.height - 400 // 공의 좌표 y
let menudx3= 7 // x속도
let menudy3 = -7// y 속도

let menux4 = canvas.width / 2 - 200// 공의 좌표 x
let menuy4 = canvas.height - 500 // 공의 좌표 y
let menudx4 = 7 // x속도
let menudy4 = -7// y 속도

let menux5 = canvas.width / 2 - 400// 공의 좌표 x
let menuy5 = canvas.height - 600 // 공의 좌표 y
let menudx5 = 7 // x속도
let menudy5 = -7// y 속도

//메뉴 셋팅 태그 변수
let div = document.getElementById("menu")
let mode;
```

추가할 기능설명
- Script 코드 설명(Game 변수)

SCript부분 코드

Game Canvas에서 사용할 변수입니다.

```
// 게임 볼 크기, 좌표, 속도 변수
let ballRadius = 20 // 볼크기
let ballRadius1 = 20 // 볼크기
let ballRadius2 = 20 // 볼크기
let x = canvas.width / 2 // 공의 좌표 x
let y = canvas.height - 200 // 공의 좌표 y
let dx = 1 // x속도
let dy = -1// y 속도

let x1= canvas.width / 2  + 100// 공의 좌표 x
let y1 = canvas.height - 600 // 공의 좌표 y
let dx1 = 1 // x속도
let dy1 = -1// y 속도

let x2 = canvas.width / 2 - 100 // 공의 좌표 x
let y2 = canvas.height - 400 // 공의 좌표 y
let dx2 = 1 // x속도
let dy2 = -1// y 속도

let ballSpeed = 2 // 공속도
let ballSpeed1 = 2 // 공속도
let ballSpeed2 = 2 // 공속도

// 볼 갯수 체크 변수
let b = true
let b1 = false
let b2 = false

// palyer 크기및 좌표 변수
const PlayerH = 20 // 플레이어 세로크기
const PlayerW = 80 // 플레이어 가로크기
let playerX = (canvas.width - PlayerW) / 2 //플레이어 x위치

// key event
let RP = false
let LP = false

//Menu 설정 변수
let start = document.getElementById("start");
let setting = document.getElementById("setting");
let bspd = document.getElementById("BallSpeed");
let bsz = document.getElementById("Ballsize");
let bc = document.getElementById("balls");

// 점수
let timerId;
let time = 0;
const score = document.getElementById("score");
let hitscore = score.innerText;
```

추가할 기능설명
- Script 코드 설명(Modevalue, Keydownhand, Keyuphand 함수)

SCript부분 코드

Modevalue 함수는 함수를 사용하는 곳의 값을 가져와 Mode 변수에 값을 저장하는 함수입니다.

Keydownhand 함수는 키보드에서 <-, ->키를 눌렀을 때 각각 RP와 LP의 값을 true로 저장하는 함수입니다.

Keyuphand 함수는 키보드에서 <-, ->키를 눌렀을 때 각각 RP와 LP의 값을 false로 저장하는 함수입니다.

```
// 점수 얻는 방법의 변수를 가져오는 함수
const modevalue = (target) =>{
    mode = target.value;
}

//키보드에서 눌렸을 때 일어나는 함수
function keydownHand(event) {
    if (event.keyCode === 39) {
        RP = true
    }
    else if (event.keyCode === 37) {
        LP = true
    }
}

//키보드에서 안 눌렸을 때 일어나는 함수
function keyupHand(event) {
    if (event.keyCode === 39) {
        RP = false
    }
    else if (event.keyCode === 37) {
        LP = false
    }
}
```

추가할 기능설명
- Script 코드 설명(Setting Event, MenuBalls 함수들)

SCript부분 코드

Id가 setting인 버튼을 클릭하면 Menu Canvas에서 입력한 공의 속도, 공의 크기, 공의 개수 값들을 각각의 변수에 저장합니다.

Menu Canvas에 사용할 각 공들을 생성하는 함수입니다.

```
// 셋팅 이벤트
setting.onclick = () =>{
    if(bspd.value != '' ){
        ballSpeed = bspd.value
        ballSpeed1 = bspd.value
        ballSpeed2 = bspd.value
    }
    if(bsz.value != '' ){
        ballRadius = bsz.value
        ballRadius1 = bsz.value
        ballRadius2 = bsz.value
    }
    if(bc.value != '' ){
        if(bc.value == 1){
            b = true
        }
        if(bc.value == 2){
            b = true
            b1 = true
        }
    if(bc.value == 3){
        b = true;
        b1 = true
        b2 = true
    }
}
```

```
//메뉴 볼
function menudrawBall1() {
    ctx.beginPath()
    ctx.arc(menux1, menuy1, menuballr1, 0, Math.PI * 2, false)
    ctx.fillStyle = "red"
    ctx.fill()
    ctx.closePath()
}
function menudrawBall2() {
    ctx.beginPath()
    ctx.arc(menux2, menuy2, menuballr2, 0, Math.PI * 2, false)
    ctx.fillStyle = "yellow"
    ctx.fill()
    ctx.closePath()
}
function menudrawBall3() {
    ctx.beginPath()
    ctx.arc(menux3, menuy3, menuballr3, 0, Math.PI * 2, false)
    ctx.fillStyle = "green"
    ctx.fill()
    ctx.closePath()
}
function menudrawBall4() {
    ctx.beginPath()
    ctx.arc(menux4, menuy4, menuballr4, 0, Math.PI * 2, false)
    ctx.fillStyle = "blue"
    ctx.fill()
    ctx.closePath()
}
function menudrawBall5() {
    ctx.beginPath()
    ctx.arc(menux5, menuy5, menuballr5, 0, Math.PI * 2, false)
    ctx.fillStyle = "purple"
    ctx.fill()
    ctx.closePath()
}
```

추가할 기능설명
- Script 코드 설명(Menudraw 함수)

SCript부분 코드

Menu Canvas를 나타내기 위한 함수입니다.

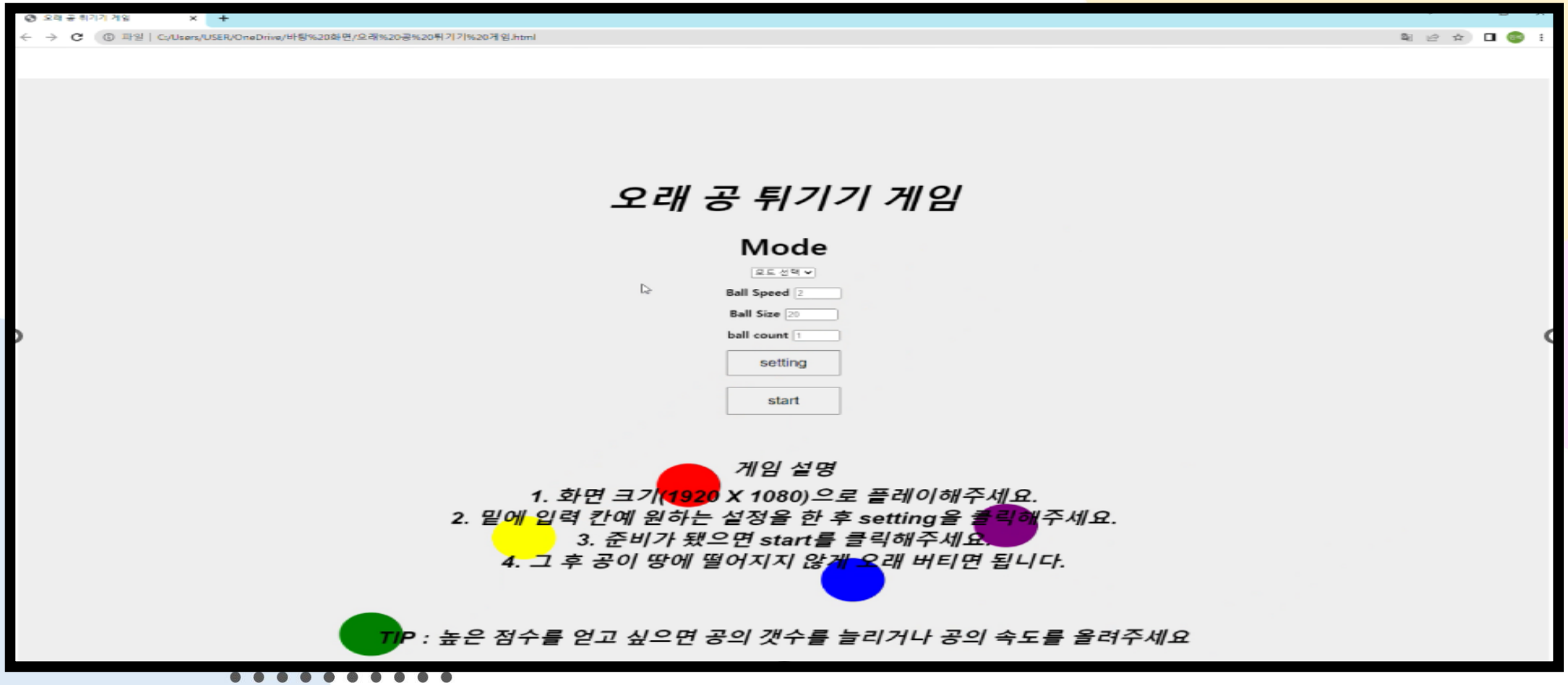
먼저 Canvas 화면을 초기화 시키고 메뉴용 공들을 만드는 함수를 호출합니다.
그다음엔 Menu Canvas에서 Display될 각 텍스트들을 삽입합니다.

마지막으로 원본 코드와 비슷하게 충돌 감지 코드와 애니메이션 효과를 넣을 Setinterval 함수를 사용합니다.

```
// 메뉴 캔버스
function manudraw(){
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    menudrawBall1();
    menudrawBall2();
    menudrawBall3();
    menudrawBall4();
    menudrawBall5();
    ctx.fillStyle = "black"; //폰트색
    ctx.textAlign = "center"; // 폰트 위치
    ctx.font = "italic bold 50px Arial, sans-serif"; // 폰트 스타일
    ctx.fillText("오래 공 튀기기 게임", canvas.width/2, canvas.height /2 - 300);
    ctx.font = "italic bold 30px Arial, sans-serif"; // 폰트 스타일
    ctx.fillText("게임 설명", canvas.width/2, canvas.height /2 + 190);
    ctx.fillText("1. 화면 크기(1920 X 1080)으로 플레이해주세요.", canvas.width/2, canvas.height /2 + 240);
    ctx.fillText("2. 밑에 입력 칸에 원하는 설정을 한 후 setting을 클릭해주세요.", canvas.width/2, canvas.height /2 + 280);
    ctx.fillText("3. 준비가 됐으면 start를 클릭해주세요.", canvas.width/2, canvas.height /2 + 320);
    ctx.fillText("4. 그 후 공이 알에 맞아지지 않게 오래 버티면 됩니다.", canvas.width/2, canvas.height /2 + 360);
    ctx.fillText("TIP : 높은 점수를 얻고 싶으면 공의 갯수를 늘리거나 공의 속도를 올려주세요", canvas.width/2, canvas.height /2 + 500);

    if(menux1 + menudx1 > canvas.width-menuballr1 || menux1 + menudx1 < menuballr1) {
        menudx1 = -menudx1;
    }
    if(menuy1 + menudy1 > canvas.height-menuballr1 || menuy1 + menudy1 < menuballr1) {
        menudy1 = -menudy1;
    }

    if(menux2 + menudx2 > canvas.width-menuballr2 || menux2 + menudx2 < menuballr2) {
        menudx2 = -menudx2;
    }
    if(menuy2 + menudy2 > canvas.height-menuballr2 || menuy2 + menudy2 < menuballr2) {
        menudy2 = -menudy2;
    }

    if(menux3 + menudx3 > canvas.width-menuballr3 || menux3 + menudx3 < menuballr3) {
        menudx3 = -menudx3;
    }
    if(menuy3 + menudy3 > canvas.height-menuballr3 || menuy3 + menudy3 < menuballr3) {
        menudy3 = -menudy3;
    }

    if(menux4 + menudx4 > canvas.width-menuballr4 || menux4 + menudx4 < menuballr4) {
        menudx4 = -menudx4;
    }
    if(menuy4 + menudy4 > canvas.height-menuballr4 || menuy4 + menudy4 < menuballr4) {
        menudy4 = -menudy4;
    }
    if(menux5 + menudx5 > canvas.width-menuballr5 || menux5 + menudx5 < menuballr5) {
        menudx5 = -menudx5;
    }
    if(menuy5 + menudy5 > canvas.height-menuballr5 || menuy5 + menudy5 < menuballr5) {
        menudy5 = -menudy5;
    }
```

```
    menux1 += menudx1;
    menuy1 += menudy1;

    menux2 += menudx2;
    menuy2 += menudy2;

    menux3 += menudx3;
    menuy3 += menudy3;

    menux4 += menudx4;
    menuy4 += menudy4;

    menux5 += menudx5;
    menuy5 += menudy5;
}
// 메뉴 시작
mpi = setInterval(manudraw, 10)
```

추가할 기능설명
- 실행 화면(Menu Canvas)



추가할 기능설명
- Script 코드 설명(Drawplayer, Drawballs 함수)

SCript부분 코드

Game Canvas에서 사용할 플레이어를 만드는 함수입니다.

Game Canvas에서 사용할 공들을 만드는 함수입니다.

```
//플레이어
function drawPlayer() {
    ctx.beginPath()
    ctx.rect(playerX, canvas.height - PlayerH, PlayerW, PlayerH)
    ctx.fillStyle = "skyblue"
    ctx.fill()
    ctx.closePath()
}
```

```
//게임 공
function drawBall() {
    ctx.beginPath()
    ctx.arc(x, y, ballRadius, 0, Math.PI * 2, false)
    ctx.fillStyle = "lightgreen"
    ctx.fill()
    ctx.closePath()
}
function drawBall1() {
    ctx.beginPath()
    ctx.arc(x1, y1, ballRadius, 0, Math.PI * 2, false)
    ctx.fillStyle = "Skyblue"
    ctx.fill()
    ctx.closePath()
}
function drawBall2() {
    ctx.beginPath()
    ctx.arc(x2, y2, ballRadius, 0, Math.PI * 2, false)
    ctx.fillStyle = "red"
    ctx.fill()
    ctx.closePath()
}
```

추가할 기능설명
- Script 코드 설명(Draw 함수)

SCript부분 코드

Game Canvas에서 시간으로 점수를 계산하는 Canvas 함수입니다.

설정 값에 따라 게임용 공을 생성하는 함수 호출과

플레이어 생성 호출하고

바닥에 충돌 했을 때 점수 상승이 멈추는 함수 호출 및

Canvas에서 점수가 안보이게 하고 clearInterval로

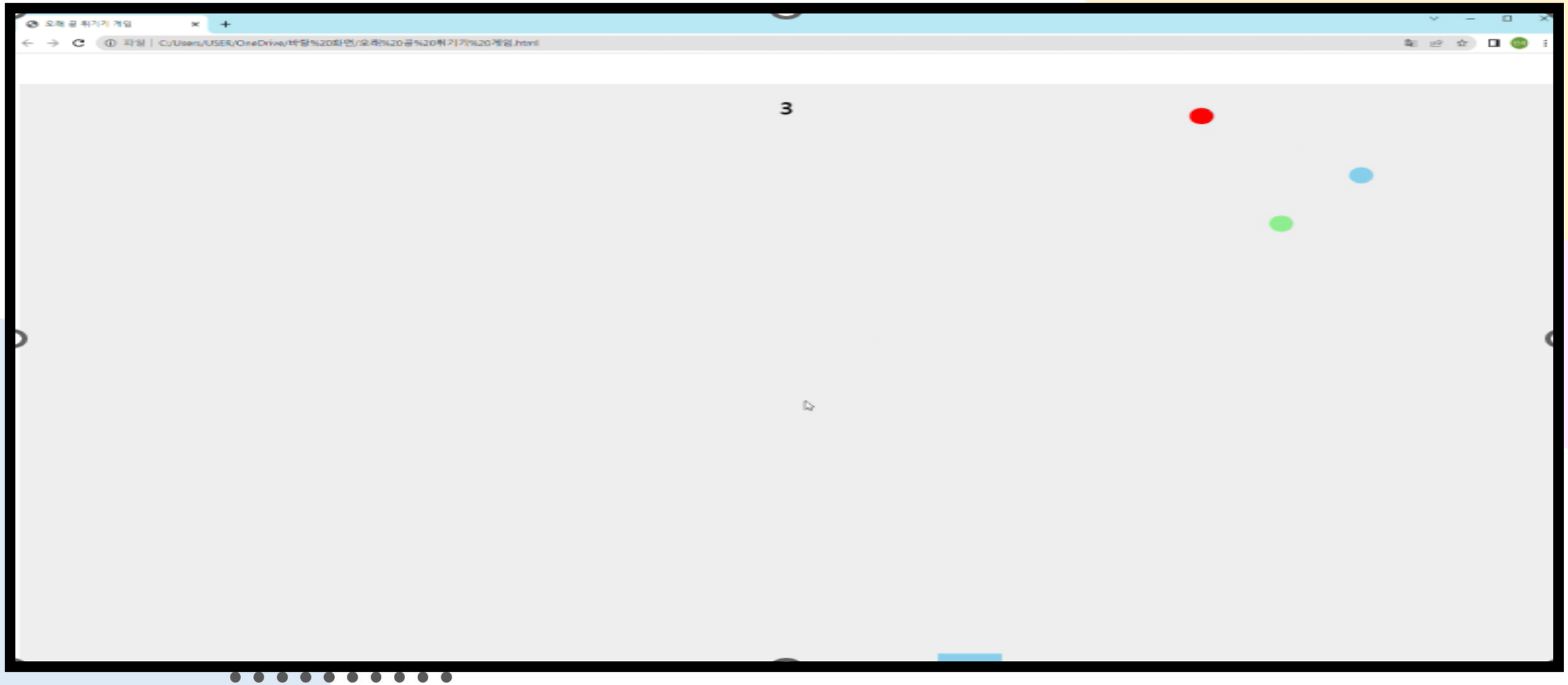
현재 Canvas 애니메이션이 종료되고

종료화면으로 이동하는 함수를 사용합니다.

```
function draw() {
    ctx.clearRect(0, 0, canvas.width, canvas.height)
    if(b == true){
        drawBall() // 공 생성
        x += dx * ballSpeed
        y += dy * ballSpeed
    }
    if(b1 == true){
        drawBall1() // 공 생성
        x1 += dx1 * ballSpeed1
        y1 += dy1 * ballSpeed1
    }
    if(b2 == true){
        drawBall2() // 공 생성
        x2 += dx2 * ballSpeed2
        y2 += dy2 * ballSpeed2
    }
    drawPlayer() // 플레이어 생성

    // 충돌 체크
    if (y + dy < ballRadius) {
        dy = -dy
    } else if (y + dy > canvas.height - ballRadius) {
        stopscore()
        score.style.display='none'
        clearInterval(tmp)
        tmp2 = setInterval(ENDdraw, 10)
    } else if (y + dy > canvas.height - ballRadius - PlayerH &&
               x + dx > playerX && x + dx < playerX + PlayerW) {
        dy = -dy
    }
    if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
        dx = -dx
    }

    if (y1 + dy1 < ballRadius1) {
        dy1 = -dy1
    } else if (y1+ dy1 > canvas.height - ballRadius1) {
        stopscore()
        score.style.display='none'
        clearInterval(tmp)
        tmp2 = setInterval(ENDdraw, 10)

    } else if (y1 + dy1 > canvas.height - ballRadius1 - PlayerH &&
               x1 + dx1 > playerX && x1 + dx1 < playerX + PlayerW) {
        dy1 = -dy1
    }
    if (x1 + dx1> canvas.width - ballRadius1 || x1 + dx1 < ballRadius1) {
        dx1 = -dx1
    }
```

```
        dy2= -dy2
    } else if (y2 + dy2 > canvas.height - ballRadius2) {
        stopscore()
        score.style.display='none'
        clearInterval(tmp)
        tmp2 = setInterval(ENDdraw, 10)
    } else if (y2 + dy2 > canvas.height - ballRadius2 - PlayerH &&
               x2 + dx2 > playerX && x2 + dx2 < playerX + PlayerW) {
        dy2 = -dy2
    }
    if (x2 + dx2 > canvas.width - ballRadius2 || x2 + dx2 < ballRadius2) {
        dx2 = -dx2
    }

    // 키 체크
    if (RP && playerX < canvas.width - PlayerW) {
        playerX += 5
    }
    if (LP && playerX > 0) {
        playerX -= 5
    }
```

추가할 기능설명
- 실행 화면(Draw Canvas)



추가할 기능설명
- Script 코드 설명(Hitdraw 함수)

SCript부분 코드

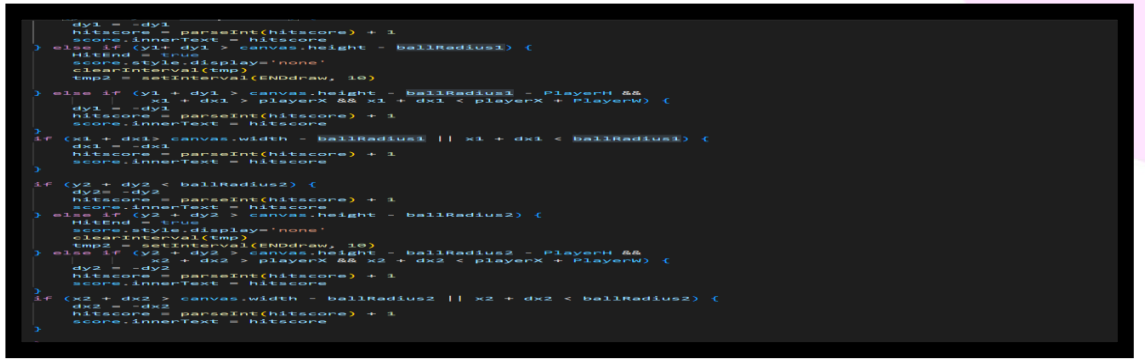
Game Canvas에서 충돌로 점수를 계산하는 Canvas 함수입니다.

기본적으로 시간으로 점수를 계산하는 Canvas와 같지만

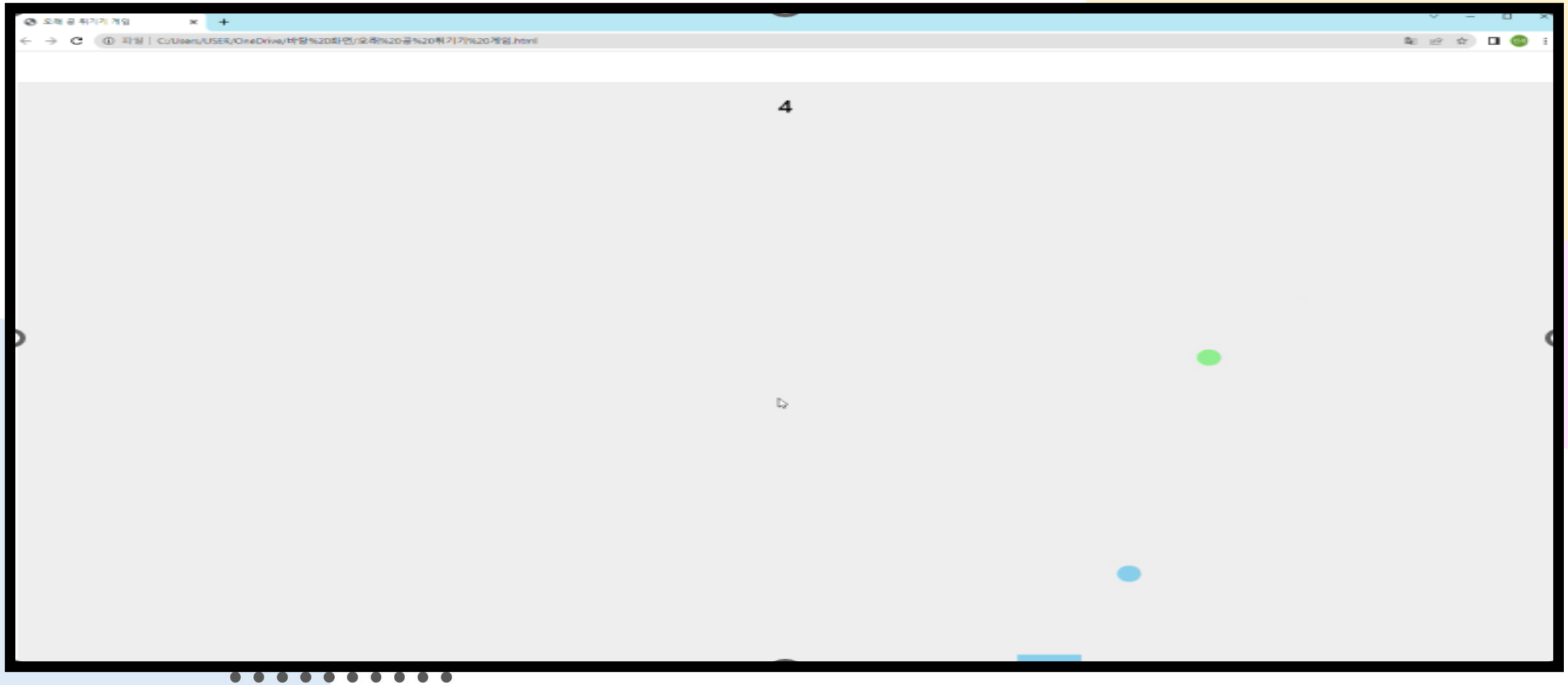
다른 점은 점수를 계산할 때 Hitscore 변수로 충돌 시 1씩 증가하도록 설정하고 공을 여러 개 하였을 때 하나의 공이 떨어져도

다른 공은 움직여서 점수를 얻는 것을 방지하기 위해 HitEnd 변수로

참이면 더 이상 점수가 오르지 않게 합니다.



추가할 기능설명
- 실행 화면(HitDraw Canvas)



추가할 기능설명
- Script 코드 설명(PrintTime, Startscore, Stopscore 함수)

SCript부분 코드

시간으로 게임하는 Canvas에서 점수에 사용하는 함수입니다.

먼저 PrintTime 함수는 time 변수가 증가되고

그 증가된 time 변수가 Id가 score인 태그에 값으로

변경시키는 함수입니다.

Startscore 함수는 PrintTime 함수를 호출하여

게임이 시작 후 값이 변경되게 하고

시작하기 전에 Setting에서 가져온 값을 가지고

각각의 공의 속도, 공의 개수에 따라 점수가 오르는 속도를 지정하며

재귀 호출을 이용해 반복 실행 되게 만든 함수입니다.

Stopscore은 게임이 끝났을 때 점수 상승이 멈출 때

사용하는 함수입니다.

```
// 점수 표현
function printTime() {
    time++;
    score.innerText = time;
}
```

```
//스코어 시작 - 재귀호출로 반복실행, 각 설정에 따른 스코어 상승속도 조절
function startscore() {
    printTime();
    if(b2 == true && b1 == true && b == true){
        if(ballSpeed == 2){
            timerId = setTimeout(startscore, 600);
        }else if(ballSpeed == 3){
            timerId = setTimeout(startscore, 566);
        }else if(ballSpeed == 4){
            timerId = setTimeout(startscore, 533);
        }else if(ballSpeed == 5){
            timerId = setTimeout(startscore, 500);
        }
    }else if(b1 == true && b == true){
        if(ballSpeed == 2){
            timerId = setTimeout(startscore, 800);
        }else if(ballSpeed == 3){
            timerId = setTimeout(startscore, 766);
        }else if(ballSpeed == 4){
            timerId = setTimeout(startscore, 733);
        }else if(ballSpeed == 5){
            timerId = setTimeout(startscore, 700);
        }
    }else if(b == true){
        if(ballSpeed == 2){
            timerId = setTimeout(startscore, 1000);
        }else if(ballSpeed == 3){
            timerId = setTimeout(startscore, 966);
        }else if(ballSpeed == 4){
            timerId = setTimeout(startscore, 933);
        }else if(ballSpeed == 5){
            timerId = setTimeout(startscore, 900);
        }
    }
}
```

```
//스코어 상승 중지
function stopscore() {
    if (timerId != null) {
        clearTimeout(timerId);
    }
}
```

추가할 기능설명
- Script 코드 설명(EndDraw 함수, 이벤트 리스너 3개)

SCript부분 코드

Enddraw 함수는 게임이 끝났을 때 Canvas를 나타내는 함수입니다.

첫 번째 이벤트 리스너는 Key를 눌렀을 때 실행이 되고

두 번째 이벤트 리스너는 Key를 땠을 때 실행이 되는 이벤트입니다.

세 번째 이벤트 리스너는 id가 start를 가진 버튼을 눌렀을 때

Setting을 설정하는 태그들을 안 보이게 만들고 메뉴 캔버스를 멈추게 하며 mode의 값에 따라 시간으로 오르는 게임 또는 충돌 시 오르는 게임을 실행하는 이벤트입니다.

```
// GAME OVER 캔버스
function ENDdraw(){
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    ctx.fillStyle = "black"; //폰트색
    ctx.textAlign = "center"; // 폰트 위치
    ctx.font = "italic bold 50px Arial, sans-serif"; // 폰트 스타일
    ctx.fillText("GAME OVER", canvas.width/2, canvas.height /2);
    ctx.font = "italic bold 35px Arial, sans-serif"; // 폰트 스타일
    ctx.fillText("SCORE : "+score.innerText, canvas.width/2, canvas.height /2 + 50);

}
```

```
// 눌린 키를 수신할 이벤트 리스너
document.addEventListener("keydown", keydownHand, false)
document.addEventListener("keyup", keyupHand, false)
start.onclick = () =>{
    div.style.display='none' // 시작 후 셋팅 메뉴 안보이게 시작
    document.getElementById('score').style.display = 'block' // 스코어 보이게
    clearInterval(tmp1) // 메뉴 캔버스 멈추기
    console.log(mode)
    if(mode == 'T'){
        startscore() // 시작 후 스코어가 화면에 나타나고 스코어 상승 시작
        tmp = setInterval(draw, 10) // 게임 캔버스 시작
    }else{
        tmp = setInterval(Hitdraw, 10) // 게임 캔버스 시작
    }

}
```

추가할 기능설명
- 추가된 기능

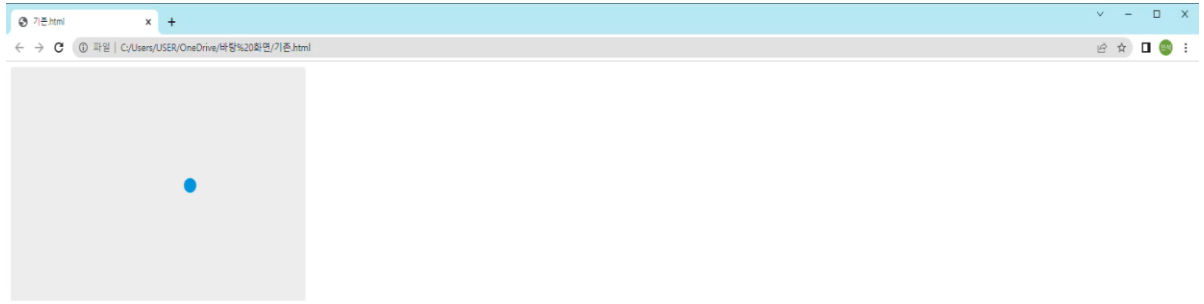
1. 타이틀 추가
2. 각 태그를 전 화면에 제약 없이 위치 조정을 할 수 있게 스타일 설정
3. 메뉴 캔버스에서 설정값들을 받기 위한 태그들 추가
4. 버튼을 누르면 설정값 가져와 각각의 설정 변수에 저장하는 기능
5. 버튼을 눌러야 게임이 시작하는 기능
6. Time을 선택하면 시간 게임, Hit를 선택하면 충돌 게임 실행하게 하는 기능
7. 플레이어를 생성하여 Canvas에 추가
8. 플레이어를 키보드로 동작하는 기능
9. 메뉴 화면 추가
10. 조건에 따라 공의 개수 및 속도 조절하는 기능
11. 시간으로 점수 계산하는 기능
12. 충돌 시 점수 계산하는 기능
13. 공이 땅에 떨어지면 게임 종료하는 기능
14. 게임 화면에 점수가 보이게 추가
15. 충돌 게임에서 공이 여러 개 일 때 종료되더라도 살아있는 다른 공의 충돌 포인트는 오르지 않게 하는 기능
16. 시간 게임에서 끝이 나면 점수가 안 오르게 하는 기능
17. 시간 게임에서 공의 개수와 속도에 따라 점수 오르는 속도 조절하는 기능
18. 게임 종료 화면 추가
19. 각 캔버스에 필요한 텍스트 추가 및 스타일설정

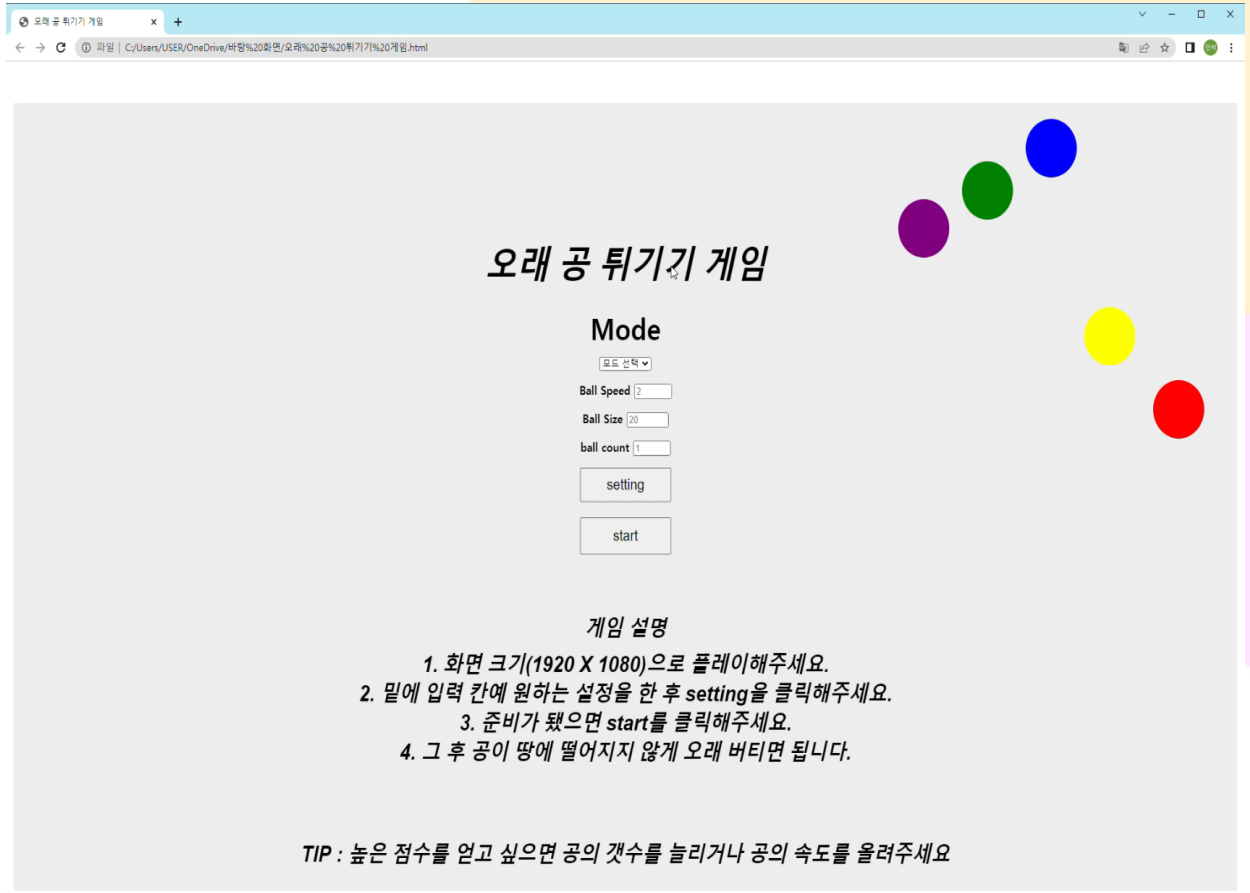
추가할 기능설명

- 개선할 기능

1. 메뉴 화면에 랭킹 버튼을 만들기
2. DB와 연동하여 랭킹 데이터를 가져오는 기능
3. 랭킹 화면 추가
4. 멀티플레이 기능
5. 화면 크기 고정

플레이 영상




오래 공 튀기기 게임
Mode
모드 선택
Ball Speed 2
Ball Size 20
ball count 1
setting
start
게임 설명
1. 화면 크기(1920 X 1080)으로 플레이해주세요.
2. 밑에 입력 칸에 원하는 설정을 한 후 setting을 클릭해주세요.
3. 준비가 됐으면 start를 클릭해주세요.
4. 그 후 공이 땅에 떨어지지 않게 오래 버티면 됩니다.
TIP : 높은 점수를 얻고 싶으면 공의 갯수를 늘리거나 공의 속도를 올려주세요

감사합니다!

출처

공 튀기기(원본 소스)

https://developer.mozilla.org/ko/docs/Games/Tutorials/2D_Breakout_game_pure_JavaScript/Bounce_off_the_walls

PPT 템플릿

https://chamssaem.com/1235